Attorney Docket No. 004006.P001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Patent Application of:

Lee McBryde, Gordon Manning, Dave Illar, Richard Williams and Michael Piszczek

Serial No. 09/882,471

Filed: June 14, 2001

For: **DATA MANAGEMENT ARCHITECTURE**

Examiner: Mujtaba M. Chaudry

Art Unit: 2133

APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
Post Office Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

Applicant submits, the following Appeal Brief pursuant to 37 C.F.R. § 41.37 for consideration by the Board of Patent Appeals and Interferences. Applicant submits payment in the amount of $500.00 to cover the cost of filing the opening brief as required by 37 C.F.R. § 41.20(b)(2). This brief does not include any new or non-admitted amendments or any new or non-admitted affidavit or other evidence.

If necessary, the Commissioner is hereby authorized in this, concurrent and future replies, to charge payment or credit any overpayment to Deposit Account No. 02-2666 for any additional fees required under 37 C.F.R. §§ 1.16 or 1.17, particularly extension of time fees.

12/09/2005 MGEBREM1 00000100 09882471
01 FC:1402 500.00 OP

I. REAL PARTY IN INTEREST

Lee McBryde, Gordon Manning, Dave Illar, Richard Williams and Michael Piszczek, the parties named in the caption, assigned their rights to that disclosed in the subject application through an assignment recorded on June 14, 2001 (011916/0515) to DataDirect Networks, Inc. of Chatsworth, California. Thus, as owner at the time the brief is being filed, DataDirect Networks, Inc. of Chatsworth, California, is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-13 are pending in the present application. Claims 1-13 are rejected. Claims 1-13 are being appealed.

IV. STATUS OF AMENDMENTS

Applicant has not amended the claims subsequent to a final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Applicant submits below a concise explanation of the subject matter defined in independent claims 1, 6, 7, 8, 12 and 13.

Applicant's claim 1 defines a data management architecture including an XOR engine 33, paragraphs [0029]-[0053], a host network interface 31, paragraphs [0027]-[0028] coupled to said XOR engine and for coupling to a host computer system, a cache 35, paragraphs [0054]-[0057] coupled to said XOR engine, a storage device interface 37, paragraphs [0058]-[0059] coupled to said cache and for coupling to a plurality of storage devices.

Applicant's claim 6 defines a data management architecture including an XOR engine 33, paragraphs [0029]-[0053], a host network interface 31, paragraphs [0027]-

[0028] coupled to said XOR engine and for coupling to a host computer system, a cache 35, paragraphs [0054]-[0057] coupled to said XOR engine, a storage device interface 37, paragraphs [0058]-[0059] coupled to said cache and for coupling to a plurality of storage devices, wherein said XOR engine includes a first transceiver 65, paragraphs [0052]-[0053] coupled to said host network interface, logic means 67, 73, 77, 69, 75, 71, paragraphs [0030]-[0052] for i) generating an XOR parity byte using said data and appending said parity byte to said data, ii) checking XOR parity, and iii) correcting detected parity errors; a second transceiver 79, paragraph [0053] coupled to said cache.

Applicant's claim 7 defines a data management architecture including an XOR engine 33, paragraphs [0029]-[0053], a host network interface 31, paragraphs [0027]-[0028] coupled to said XOR engine and for coupling to a host computer system, a cache 35, paragraphs [0054]-[0057] coupled to said XOR engine, said cache including a plurality of cache segments 85, paragraphs [0054]-[0057], each of said cache segments including a dual port memory array 95, paragraph [0055]-[0057], a bus expander 91, paragraphs [0055]-[0057] coupled between said XOR engine and said dual port memory array, a bus funnel 97, paragraphs [0055]-[0057] coupled between said XOR engine and said dual port memory array, and a buffer 99, paragraphs [0055]-[0057] coupled between a storage device interface 37, paragraphs [0058]-[0059] and said dual port memory, said storage device interface 37, paragraphs [0058]-[0059] coupled to said cache and for coupling to a plurality of storage devices, wherein said XOR engine includes a first transceiver 65, paragraphs [0052]-[0053] coupled to said host network interface, logic means 67, 73, 77, 69, 75, 71, paragraphs [0030]-[0052] for i) generating in real time an XOR parity byte using said data and appending said parity byte to said data, ii) checking in real time XOR parity, and iii) correcting in real time detected parity errors, a second transceiver 79, paragraph [0053] coupled to said cache.

Applicant's claim 8 defines a data management architecture including an XOR engine 33, paragraphs [0029]-[0053] which operates to generate in real time an XOR parity byte using said data and append said parity byte to said data, ii) check in real time XOR parity, and iii) correct in real time detected parity errors, a host network

interface 31, paragraphs [0027]-[0028] coupled to said XOR engine and for coupling to a host computer system, a cache 35, paragraphs [0054]-[0057] coupled to said XOR engine, a storage device interface 37, paragraphs [0058]-[0059] coupled to said cache and for coupling to a plurality of storage devices.

Applicant's claim 12 defines a data management architecture including an XOR engine 33, paragraphs [0029]-[0053], a host network interface 31, paragraphs [0027]-[0028] coupled to said XOR engine and for coupling to a host computer system, a cache 35, paragraphs [0054]-[0057] coupled to said XOR engine, a storage device interface 37, paragraphs [0058]-[0059] coupled to said cache and for coupling to a plurality of storage devices, wherein said XOR engine includes, a first transceiver 65, paragraphs [0052]-[0053] coupled to said host network interface, logic means 67, 73, 77, 69, 75, 71, paragraphs [0030]-[0052] for i) generating in real time an XOR parity byte using said data and appending said parity byte to said data, ii) checking in real time XOR parity, and iii) correcting in real time detected parity errors, a second transceiver 79, paragraph [0053] coupled to said cache.

Applicant's claim 13 defines a data management architecture including an XOR engine 33, paragraphs [0029]-[0053], a host network interface 31, paragraphs [0027]-[0028] coupled to said XOR engine and for coupling to a host computer system, a cache 35, paragraphs [0054]-[0057] coupled to said XOR engine, said cache including a plurality of cache segments 85, paragraphs [0054]-[0057], each of said cache segments including i) a dual port memory array 95, paragraph [0055]-[0057], ii) a bus expander 91, paragraphs [0055]-[0057] coupled between said XOR engine and said dual port memory array, iii) a bus funnel 97, paragraphs [0055]-[0057] coupled between said XOR engine and said dual port memory array, and iv) a buffer 99, paragraphs [0055]-[0057]coupled between said storage device interface and said dual port memory, a storage device interface 37, paragraphs [0058]-[0059] coupled to said cache and for coupling to a plurality of storage devices, wherein said XOR engine includes, a first transceiver 65, paragraphs [0052]-[0053] coupled to said host network interface, logic means 67, 73, 77,

69, 75, 71, paragraphs [0030]-[0052] for i) generating in real time an XOR parity byte using said data and appending said parity byte to said data, ii) checking in real time XOR parity, and iii) correcting in real time detected parity errors, a second transceiver 79, paragraph [0053] coupled to said cache.

The logic means of Claims 2, 5, 7, 12 and 13 is shown in Figure 6 and described at paragraphs [0030]-[0052] as comprising parity replacement mux 67, transmit XOR 73, receive XOR 77, XOR regen 69, parity error detector 75, and lane mux 71.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claim 1 is anticipated under 35 U.S.C. §102(b) by U.S. Patent No. 5,883,909 issued to DeKoning et al. ("DeKoning").

Whether claims 2-13 are unpatentable under 35 U.S.C. §103(a) by DeKoning in view of U.S. Patent No. 5,668,971 issued to Neufled ("Neufled").

Applicant presents the above-mentioned grounds of rejection for review.

VII. ARGUMENT

A. It is asserted in the Office Action that Claim 1 is rejected under 35 U.S.C. §102(b) as being anticipated by DeKoning.

Claim 1

The essence of the invention as explained in Applicant's specification at page 2, is that in a RAID controller, placing the parity XOR engine on the host network side of the data cache maximizes data transfer bandwidth with minimum parity calculation overhead. This aspect of the invention, as defined in Claim 1 and shown, in Figure 3, allows for the use of pipelined register sets by which the XOR engine can calculate, check and correct any errors in real time during data transfers.

Summary of Argument re Claim 1 Rejection:

The following discussion sets forth Applicant's analysis with respect to the patentability of claim 1.

It is asserted in the July 28, 2003 Office Action at page 2, that DeKoning teaches:

"an XOR engine (62) in Figure 2 . . .

a host network interface (16) coupled to the XOR engine, which is inside the RPA (20) for coupling to a host computer system in Figure 1. . .
a cache (64) coupled to the XOR engine in Figure 2. . .
a storage device (22) interface coupled to the cache (64, Fig. 2), which is inside the RPA (20) in Figure 1."

It is noted that DeKoning el al. defines RPA as RAID Parity Assist (col 4:5-6).

In a subsequent Office Action dated December 16, 2004, in seeking to clarify DeKoning's teachings, the Examiner states:

> "In particular, DeKoning et al., teaches (Figures 1 and 2) the XOR engine (62) coupled to the memory controller (60), which is analogous to the host/network interface of the present application. The memory controller (60) is also coupled to the PCI bus (28)."

That is, the Examiner contends that DeKoning's memory controller 60 is analogous to Applicant's host/network interface 33. It is Applicant's position that this contention by the Examiner is in error for the following reasons.

Details of Argument re Claim 1 Rejection:

The host/network interface of the present invention is shown in block diagram form in Figure 5 with corresponding description at pages 4 and 5. For example, at page 4, Applicant defines host/network interface 33 as:

> "a communications interface to a host computer or a network of computers. In one embodiment, the invention maintains a ANSI-X3T11 fiber channel interface utilizing a SCSI command set on the front end . . ."

RPA memory controller 60 of DeKoning is not analogous to Applicant's host/network interface, i.e., a communications interface such as a fiber channel interface using a SCSI command set on the front end. In particular, DeKoning clearly shows RPA memory controller 60 connected to system PCI bus 28 and not for coupling to a host computer system as claimed by Applicant. Although not shown in Figure 2, it is apparent that PCI bus 28 is the same PCI bus 28 shown in Figure 1. For example, in column 4, line 42, it is stated "The RPA memory controller 60 controls (1) the flow of data between the system bus 28, the RPA memory 22, and the intermediate parity buffer 64, and (2) the operation of the XOR engine 62." Applicant submits that a person skilled in the art would not construe a memory controller as analogous to a host/network interface in view of the completely different functions performed. Furthermore, it is apparent that Figure 2 is not a complete system diagram, but rather the elements shown above bus 28 in Figure 1. That is, the host interface 16 of Figure 1 also exists in the system of Figure 2

even though not shown. Thus, to the extent that there is an analogous element in DeKoning to Applicant's host/network interface, it is DeKoning's host interface 16 which is shown as being connected to host device 3. For example, in Column 6 of DeKoning, it is stated at line 26:

> "chunk 4 is transferred from the host device 31, through a host interface circuit 16, across the system bus 28, through the RPA memory controller 60."

Clearly, DeKoning's RPA memory controller 60 is not analogous to DeKoning's host interface 16.

In DeKoning, XOR engine 62 as shown in Figure 2 is connected to RPA memory controller 60. Engine 61 and controller 60 are inside RPA 20. RPA memory controller 60 then, through PCI bus 28, is connected to host interface 16. As described in DeKoning at column 4 beginning at line 45:

> "the XOR engine 62 primarily performs a bit-wise XOR operation on data stored in the intermediate parity buffer 64 with data received from the RPA memory controller 60 . . . The RPA memory 22 permits the intermediate storage of blocks of read data (i.e., data read from the disk array 33 which is sent to the host device 31) and blocks of write data (i.e., data received from the host device 31 which is written to the disk array 33). As a result, the RPA memory 22 provides a caching function wherein the disk array controller 10 may utilize read/write data stored in the RPA memory 22 instead of accessing one or more of the disk drives associated with the disk array 33."

By way of contrast, using the XOR engine of the present invention coupled to the host network interface, the XOR engine receives data from the host network interface via a transceiver (65 shown in Figure 6) so that during a host write/data transfer, the XOR engine can calculate the parity by XORing the data bits from the host network interface. There is no RPA memory controller 60 or equivalent element between the host network interface and the XOR engine as required by DeKoning. The differences between the present invention and DeKoning should be apparent with reference to Figure 2 of DeKoning which shows that XOR engine 62 is connected to RPA memory controller 60 on one side and buffer 64 on the other side with **no** connection to the host interface 16 (other than through memory controller 60) or to the storage device interface (other than through memory controller 60). Clearly, memory controller 60 which, as described in DeKoning at column 4, beginning at line 42, controls the flow of data between the system bus, the RPA memory and the intermediate parity buffer and the operation of the XOR engine 62. Thus, memory controller 60 is not a passive device between the XOR engine and the host network interface, but is a required and active element resulting in an entirely different architecture from that described and claimed by Applicant. While data from

XOR 62 presumably can flow to and from host interface 16 in DeKoning, the XOR engine 62 in DeKoning is not coupled to the host interface 16 as disclosed and claimed by Applicant due to the existence of the RPA memory controller 60 between the XOR engine and the host network interface.

Therefore, since DeKoning does not disclose, teach or suggest an XOR engine coupled to a host network interface, it does not disclose all of Applicant's claim 1 limitations. Therefore, Applicant respectfully asserts that a *prima facie* rejection under 35 U.S.C. § 102(b) has not been adequately set forth relative to DeKoning. Thus, Applicant's claim 1 is not anticipated by DeKoning.

Accordingly, withdrawal of the 35 U.S.C. § 102(b) rejection of claim 1 is respectfully requested. Further, Applicant notes that all of the claims contain a host network interface coupled to the XOR engine. For the reasons noted above, DeKoning does not provide this teaching. Therefore, unless the other prior art cited by the Examiner contains this teaching, the remaining claims cannot be deemed obvious if DeKoning is relied upon for this teaching.

B. It is asserted in the Office Action that Claims 2-13 are rejected under 35 U.S.C. §103(a) as being unpatentable under 35 U.S.C. §103(a) over DeKoning in view of Neufled. The following discussion sets forth in detail Applicant's analysis with respect to the patentability of claims 2-13.

1. Claim 2, 4, 6 and 7

Claim 2 is directed to a specific implementation of an XOR engine as illustrated in Figure 6. The Examiner relies upon Neufeld for its alleged teaching of first and second transceivers of the type set forth in Claim 2 in the July 28, 2003 Office Action. Although Neufeld discloses transceivers 320 and 319 in Figure 9, there is no logic means which operates on data from the first transceiver as disclosed by Applicant and as defined in Claim 2. Neufled discloses an XOR engine in the Background of the Invention, but there is no connection between that prior art XOR engine and the transceivers 319 and 329 shown in Figure 9 of Neufled.

In response to Applicant pointing out that the logic means of Claim 2 generates an XOR parity byte, checks XOR parity and corrects detected parity errors, in the December 16, 2004 Office Action, the Examiner contends at page 3 that Neufeld teaches a dedicated XOR engine

which generates parity data on a word for word basis from up to four different transfer blocks and is capable of writing the result to a specified drive or to a transfer offered through a sub-channel. The Examiner also contends that Neufeld's trap logic 300 corresponds to the claimed logic means. However, such teaching is irrelevant since there is no teaching in Neufled of detecting and correcting parity errors. Thus, Neufled cannot be relied upon as a basis for rejecting claim 2.

Moreover, Neufeld does not provide the necessary teachings missing from DeKoning relative to the coupling of the XOR engine to the host network interface as explained above.

Claim 4 details the construction of the cache as comprising a plurality of cache segments. Applicant does not assert any separate patentability as to Claim 4. Claim 6 is essentially Claim 2 rewritten in independent form. Claim 7 is similar to claims 2 and 4.

2. Claims 3 and 5

Claim 3 is directed to a specific implementation of a host/network interface as illustrated in Figure 5. In support of the rejection of Claim 3, the Examiner contends that DeKoning teaches a physical interface (10), a protocol engine (20) and a memory (22) as shown in Figure 1. The Examiner relies upon Neufled to supply a microcontroller (62) and interface buffers (60) as shown in Figure 1 and receive/transmit buffers 316, 319 and 320 shown in Figure 9. In response, Applicant notes that DeKoning defines element 10 as a disk array controller (1:37) and element 20 as a RAID Parity Assist circuit (1:40). Applicant's physical interface 51 is defined as a transceiver (paragraph [0028], line 3. Clearly, a transceiver is not a disk array controller. Applicant's protocol engine is described as an LSI Merlin™ Fibre Channel core which is described at http://www.lsilogic.com/files/docs/marketing_docs/coreware/final_gb_pb.pdf. Clearly, such a protocol engine is not a RAID Parity Assist circuit of the type disclosed by DeKoning. Thus, even assuming that Neufled teaches the missing microcontroller and buffers, the combination would still not result in an interface as defined by Claim 3. Noteworthy also is the fact that Nuefled's element 62 is not a microcontroller as alleged by the Examiner but is actually a memory management unit (col 9:9) which is part of a larger circuit which does not appear to in any way relate to a host network interface as shown in Figure 5 and defined in Claim 3. The Examiner appears to have taken isolated elements such as buffers, microcontrollers and the like from the prior art and somehow concluded that since such elements exist, and without regard to their interoperability with each other, Applicant's specific implementation is obvious.

Of course, if this were the standard for patentability, no invention which consists solely of prior art elements would be entitled to patent protection.

Claim 5 is directed to a specific implementation of a storage device interface as illustrated in Figure 9. In support of the rejection of Claim 5, the Examiner contends that DeKoning teaches a physical interface (10), a protocol engine (20) and a memory (22) as shown in Figure 1. The Examiner relies upon Neufled to supply a microcontroller (62) and interface buffers (60) as shown in Figure 1 and receive/transmit buffers 316, 319 and 320 shown in Figure 9. Applicant notes that since the rejection of Claim 5 is based on the identical reasons set forth for Claim 3, the rejection of Claim 5 cannot be maintained for the same reasons.

3. Claims 8-13

Claim 8-13 are similar to Claims 1 and 3-7 respectively but each adds the limitation that the XOR engine operates in real time to generate an XOR parity byte, check in real time XOR parity and correct in real time detected parity errors. In this connection, Applicant submits that the term real time is a well defined term in the computer arts which provides operational advantages of systems which do not operate in real time. The Examiner in an Office Action dated December 16, 2004 asserted the same grounds for rejection of Claims 8-13 as Claims 1-7 with no attempt made to take into account the added limitation of the XOR engine operating to generate in real time an XOR parity byte. Thus, since the Examiner has not cited any structure (or even alleged) that the prior art of record teaches or even suggests real time operations, the rejection of claims 8-13 on the same basis as claims 1 and 3-7 cannot be sustained. Applicant notes that neither DeKoning nor Neufled even mentions the term real time in any context. Accordingly, Applicant submits that neither reference can be relied upon as a basis for rejecting claims 8-13.

CONCLUSION

Based on the foregoing, Applicant requests that the Board overturn the rejection of all pending claims and hold that all of the claims of the present application are allowable.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR, & ZAFMAN LLP

Dated: December 6, 2005

By: _____
Eric S. Hyman, Reg. No. 30,139

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(310) 207-3800

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited as First Class Mail, with sufficient postage, with the United States Postal Service in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, Post Office Box 1450, Alexandria, Virginia 22313-1450 on December 6, 2005.

_____ 12/6/05
Kumiko Alexander

VIII. CLAIMS APPENDIX

The claims involved in this Appeal are as follows:

1. (Original) A data management architecture comprising:

a) an XOR engine;

b) a host network interface coupled to said XOR engine and for coupling to a host computer system;

c) a cache coupled to said XOR engine;

d) a storage device interface coupled to said cache and for coupling to a plurality of storage devices.

2. (Original) The data management architecture defined by Claim 1 wherein said XOR engine comprises:

a) a first transceiver coupled to said host network interface;

b) logic means for i) generating an XOR parity byte using said data and appending said parity byte to said data, ii) checking XOR parity, and iii) correcting detected parity errors;

c) a second transceiver coupled to said cache.

3. (Original) The data management architecture defined by Claim 1 wherein said host network interface comprises:

a) a physical interface;

b) a protocol engine coupled to the physical interface;

c) a receive buffer coupled to the protocol engine;

d) a transmit buffer coupled to the protocol engine;

e) interface buffers coupled to the transmit and receive buffers;

f) a bus coupled to the protocol engine;

g) a microcontroller coupled to the bus;

h) a memory coupled to the bus.

4. (Original) The data management architecture defined by Claim 1 wherein said cache comprises:

a plurality of cache segments, each of said cache segments including i) a dual port memory array, ii) a bus expander coupled between said XOR engine and said dual port memory array, iii) a bus funnel coupled between said XOR engine and said dual port memory array, and iv) a buffer coupled between said storage device interface and said dual port memory.

5. (Original) The data management architecture defined by Claim 1 wherein said storage device interface comprises:

a) a physical interface;

b) a protocol engine coupled to the physical interface;

c) a receive buffer coupled to the protocol engine;

d) a transmit buffer coupled to the protocol engine;

e) interface buffers coupled to the transmit and receive buffers;

f) a bus coupled to the protocol engine;

g) a microcontroller coupled to the bus;

h) a memory coupled to the bus.

6. (Original) A data management architecture comprising:

a) an XOR engine;

b) a host network interface coupled to said XOR engine and for coupling to a host computer system;

c) a cache coupled to said XOR engine;

d) a storage device interface coupled to said cache and for coupling to a plurality of storage devices,

wherein said XOR engine includes:

a first transceiver coupled to said host network interface;

logic means for i) generating an XOR parity byte using said data and appending said parity byte to said data, ii) checking XOR parity, and iii) correcting detected parity errors;

a second transceiver coupled to said cache.

7. (Original) A data management architecture comprising:

a) an XOR engine;

b) a host network interface coupled to said XOR engine and for coupling to a host computer system;

c) a cache coupled to said XOR engine, said cache including a plurality of cache segments, each of said cache segments including i) a dual port memory array, ii) a bus expander coupled between said XOR engine and said dual port memory array, iii) a bus funnel coupled between said XOR engine and said dual port memory array, and iv) a buffer coupled between said storage device interface and said dual port memory;

d) a storage device interface coupled to said cache and for coupling to a plurality of storage devices,

wherein said XOR engine includes:

a first transceiver coupled to said host network interface;

logic means for i) generating an XOR parity byte using said data and appending said parity byte to said data, ii) checking XOR parity, and iii) correcting in detected parity errors;

a second transceiver coupled to said cache.

8. (Previously Presented) A data management architecture comprising:

a) an XOR engine which operates to generate in real time an XOR parity byte using said data and append said parity byte to said data, ii) check in real time XOR parity, and iii) correct in real time detected parity errors;

b) a host network interface coupled to said XOR engine and for coupling to a host computer system;

c) a cache coupled to said XOR engine;

d) a storage device interface coupled to said cache and for coupling to a plurality of storage devices.

9. (Previously Presented) The data management architecture defined by Claim 8 wherein said host network interface comprises:

a) a physical interface;

b) a protocol engine coupled to the physical interface;

c) a receive buffer coupled to the protocol engine;

d) a transmit buffer coupled to the protocol engine;

e) interface buffers coupled to the transmit and receive buffers; a bus coupled to the protocol engine;

g) a microcontroller coupled to the bus;

h) a memory coupled to the bus.

10. (Previously Presented) The data management architecture defined by Claim 8 wherein said cache comprises:

a plurality of cache segments, each of said cache segments including i) a dual port memory array, ii) a bus expander coupled between said XOR engine and said dual port memory array, iii) a bus funnel coupled between said XOR engine and said dual port memory array, and iv) a buffer coupled between said storage device interface and said dual port memory.

11. (Previously Presented) The data management architecture defined by Claim 8 wherein said storage device interface comprises:

a) a physical interface;

b) a protocol engine coupled to the physical interface;

c) a receive buffer coupled to the protocol engine;

d) a transmit buffer coupled to the protocol engine;

e) interface buffers coupled to the transmit and receive buffers;

f) a bus coupled to the protocol engine;

g) a microcontroller coupled to the bus;

h) a memory coupled to the bus.

12. (Previously Presented) A data management architecture comprising:

a) an XOR engine;

b) a host network interface coupled to said XOR engine and for coupling to a host computer system;

c) a cache coupled to said XOR engine;

d) a storage device interface coupled to said cache and for coupling to a plurality of storage devices,

wherein said XOR engine includes:

a first transceiver coupled to said host network interface;

logic means for i) generating in real time an XOR parity byte using said data and appending said parity byte to said data, ii) checking in real time XOR parity, and iii) correcting in real time detected parity errors;

a second transceiver coupled to said cache.

13. (Previously Presented) A data management architecture comprising:

a) an XOR engine;

b) a host network interface coupled to said XOR engine and for coupling to a host computer system;

c) a cache coupled to said XOR engine, said cache including a plurality of cache segments, each of said cache segments including i) a dual port memory array, ii) a bus expander coupled between said XOR engine and said dual port memory array, iii) a bus funnel coupled between said XOR engine and said dual port memory array, and iv) a buffer coupled between said storage device interface and said dual port memory;

d) a storage device interface coupled to said cache and for coupling to a plurality of storage devices,

wherein said XOR engine includes:

a first transceiver coupled to said host network interface;

logic means for i) generating in real time an XOR parity byte using said data and appending said parity byte to said data, ii) checking in real time XOR parity, and iii) correcting in real time detected parity errors;

a second transceiver coupled to said cache.

IX. EVIDENCE APPENDIX

Applicant does not submit further evidence as the evidence relied on for the grounds of rejection pertain only to the cited prior art.

X. RELATED PROCEEDINGS APPENDIX

Applicant asserts there are no related proceedings that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.